Robot Skill Transfer Based on B-Spline Fuzzy Controllers For Force-Control Tasks

Markus Ferch, Jianwei Zhang and Alois Knoll
Technical Computer Science, Faculty of Technology,
University of Bielefeld, 33501 Bielefeld, Germany

Abstract

Human–beings can easily describe their behaviour by `IF-THEN` *rules, which can be transferred from one task to another with slight local changes. However, standard techniques for function approximation like neuronal networks or associative memories are unable to work with rules. We introduce a method for extracting and importing human readable rules from and to a B-spline fuzzy controller. Rule import is used to initialise a B-spline fuzzy controller with a priori knowledge to decrease the learning time and overcome the problem of partially trained B-spline controllers. In the experimental section we show how a set of rules for a two arm cooperation task are generated through "learning-by-doing" and transferred to a robot screwing operation. The successful experiment shows how rule-based knowledge can be used for skill transfer in similar tasks.*

1 Introduction

Like conventional process control, perception-action cycles can be implemented with either model-based or connectionist methods. Model-based approaches must specify explicit sensor-robot system models. Typical applications are calibrated methods for hand-eye coordination and the artificial potential-field for collision-avoidance. However, they suffer from the following problems:

- They are not adaptable to varying environments;
- They cannot be built incrementally or modularly;
- They cannot be interpreted symbolically.

Connectionist approaches use expert knowledge or learning to acquire the characteristics of the sensor-action system. Recently, such approaches were applied to the sensor-based control of robots, as well as to classical process control. Applications of artificial neural networks [1, 2, 3, 4] exhibit intelligent behaviour like self-organisation, adaptation and distributed processing but the "black-box" structure remains as an obstacle to integrating symbolic approaches which represent the other important part of human intelligence. Fuzzy control also finds applications in behaviour implementation [5], but these controllers are mainly realised by human expert knowledge instead of self-adaptation.

One important aspect of human intelligence for control is the rapid understanding and application of experiences told by others. Based on this capability, a human does not need to learn every control task from the very beginning. Fuzzy control in combination with adaptive learning can provide a suitable tool for obtaining similar transfer abilities in machine control. Let us show the transfer procedure for a control task with a simple one-dimensional example. Consider a control system with one sensor input x and one action output. Consider that the output should react to the input like a $\sin(2\pi x^2)$ function. Assume that a controller A has learned through (a lot of) effort how to achieve the optimal control (the best approximation through supervised learning), Fig. 1. The controller can extract the following set of symbolic rules interpreting the controller behaviour:

```
IF Input IS zero    THEN Action IS zero
IF Input IS small   THEN Action IS positive_mid
IF Input IS medium  THEN Action IS positive_big
IF Input IS large   THEN Action IS negative_big
IF Input IS maximum THEN Action IS zero
```

If controller A tells controller B its control experiences qualitatively, these coarse control rules can be applied. Although controller B can possibly have its own interpretation of the linguistic terms, which means slightly different membership functions, it can still achieve similar control results without any learning process.

In this paper we discuss the usage of rule-based knowledge with a B-spline fuzzy controller [6] and the transfer between two similar control tasks. Such a skill transfer is cognitively adequate and leads to faster convergence of learning processes. Focussing on the two aspects of rule extraction and rule import, we describe the learned data as `IF-THEN` rules, and in a second step, how to initialise a controller by *a priori* known rules. The methods are tested by composing both algorithms and comparing the results. We also designed a classical fuzzy controller based on the extracted rules. We have used a two manipulator cooperation task [6] and a screwing operation [7] to show how

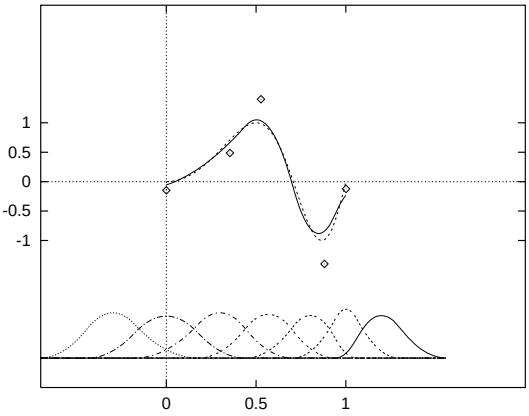

Figure 1: Learning to map the sensor readings into the action values for emulating the function $y = \sin(2\pi x^2)$. The B-spline basis functions defined on the interval $[0, 1]$ represent the linguistic terms “zero”, “small”, “medium”, “large”, “maximum” (from left to right).The values of the rhombi represent the linguistic terms of the control action, “zero”, “positive_medium”, “positive_large”, “negative_large”, “zero”.

rule-based knowledge can be transferred from one task to another. The learning time can be reduced significantly and extremal values in the learning process like force spikes can be avoided.

2 Basic Methods

2.1 Rule-Based Controllers

A sample rule for compliant motion could be:

```
IF the deviation from the desired force is very
                        high
  THEN the arm should move back a big stretch
```

A general control rule has the following form:

$$\texttt{IF Input}_1\texttt{ is PB AND ... AND Input}_N\texttt{ is NB THEN Output is ZO (0.9754)}$$

where $\texttt{Input}_n$ is the n-th input dimension, `PB, PM, PS, ...` represent linguistic terms like Positive Big or Positive Medium ... and the value in the brace is a rule weight.

2.2 Rule Extraction

Assuming a trained B-spline controller for a given problem/function, we want to describe the output dimension with triangular membership functions. We have to find an optimal set of linguistic terms which matches the output of the premises. Especially for higher dimensional problems this can be realised by a clustering method (e.g. fuzzy c-means) where the cluster centres are associated with the maxima of the triangular linguistic terms.

In the rule generation process all input combinations related to the B-splines are generated. Therefore we use the position of the B-spline extremal points. For third–order B-splines the maximum of the i-th spline can be computed by:

$$x_{\max} = \frac{-\,x_{i+2}\cdot x_{i+3} + x_i \cdot x_{i+1}}{-x_{i+3} + x_{i+1} - x_{i+2} + x_i}$$

where x_i are the knot values of the related knotvector with which the B-spline is defined. When the controller output at this point is untrained, no rule is generated. Otherwise the output membership function with the highest degree of membership d is searched, where d is used as a rule weight. If d is less then 1.0, the controller output intersects with two linguistic terms of the output dimension, and therefore the local control action cannot be described exactly by one rule. Another rule with the same premise but the other intersected linguistic term as conclusion has to be generated.

This gives us a basic approach for rule reduction. Under the loss of accuracy we can specify a limit $b < 1$. When the degree of membership d is below the limit ($d < b$), a second rule is generated.

The more output membership functions are used, the better these linguistic terms are matched and the less likely we need a second rule to describe the real output value. These leads to a rule description in zero-order TSK type whose output values are crisp.

$$\texttt{IF Input}_1\texttt{ is PB AND ... AND Input}_N\texttt{ is NB THEN Output is Value}$$

where `Value` is an output value, based on which a perfect reconstruction can be realised by a minimum number of rules. The singletons are usefull for exact function approximation but problematic for the understanding of the rules, therefore we propose to use linguistic terms and accept the loss of accuracy in function approximation.

2.3 Rule Import

The import process can be understood as a learning process for the B-spline fuzzy controller. The given rules are input-output combinations that have to be learned. The error is the difference between the conclusion and the actual controller output $f(\vec{x}_{j_1 \ldots j_N})$ on the corresponding input values multiplied by the rule weight.

Therefore, an algorithm generates the input values from the premise for every rule and calculates the actual controller

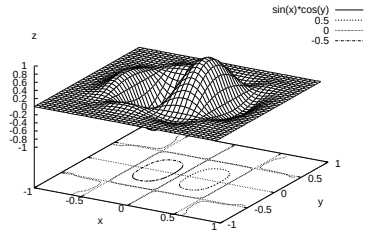

Figure 2: $sin(x) \cdot cos(y)$ learned by a B-spline controller in ranges of $[-1, 1]$.

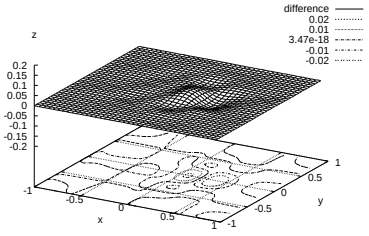

Figure 3: Difference surface between the original Fig. 2 and the imported function (b=1).

output. If the difference to the conclusion is under a certain limit, no learning step is performed. Otherwise this difference is used as the learning error. The rule weight is used as the learning step rate. This process is repeated until the mean–squared error over all learning errors remains constant. Because this is a gradient descent of a quadratic error function, the convergence can be guaranteed. Fig. 2, 3 and 4 show the composition of extraction and import of a two dimensional function $(sin(x) \cdot cos(y))$. If we use a limit $b = 0.8$ to reduce the number of rules, the reconstruction of the original function is not perfect, as can be seen in Fig. 4. The function structure can also be generated by a classical Mamdani fuzzy controller but it is not very smooth (Fig. 5).

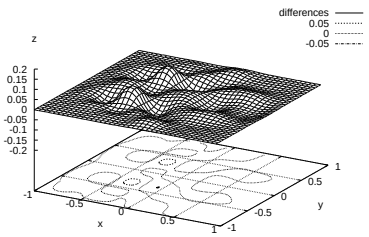

Figure 4: Difference surface if the rules are reduced as described above $b = 0.8$.

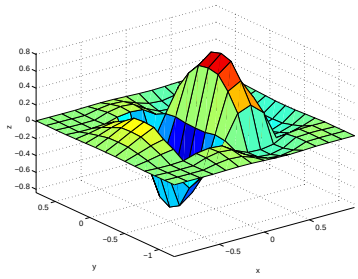

Figure 5: Feeding a Mamdani type fuzzy controller with the extracted rules.

3 Experiments for Force–Control Tasks

We now discuss how to transfer the rules which are learned by one task to another analogous problem. The first domain is a two-arm cooperation task [6] (Fig. 6), the second one is a robot screwing operation [7] (Fig. 7).

3.1 Two-Arm Transportation Skill

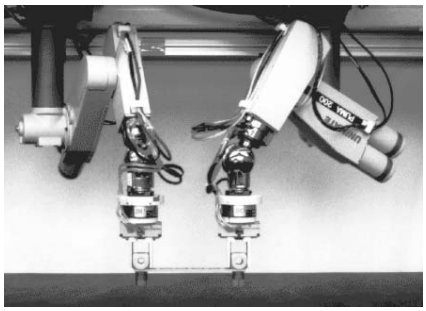

Figure 6: Two-arm transportation task of a wooden ledge.

In the cooperation task two robots rigidly grasp an object and move it along a desired trajectory. Both manipulators are coupled in a single kinematic chain. The resulting forces are measured with two Force Torque Sensors (FTS) and are fed back via a B-spline controller to compute positional corrections. The task is realised by three controllers for the forces in x, y, z direction and one controller for the torque in z direction.

3.2 Screwing Skill

In the screwing scenario one robot is holding a bolt while another robot is performing the screwing, Fig. 9. The complete operation can be decomposed in the following steps: *Find the contact, put the bolt in the nut, find the notch point, screw-in, terminate (when one hand detects a torque impulse in the approach direction).* We focus on the *screw-in* part of this sequence which is somehow similar to the cooperation task. Three controllers are used to adapt the forces in x, y, z direction. Only the position of one robot needs to be changed which allows us to skip the drifting

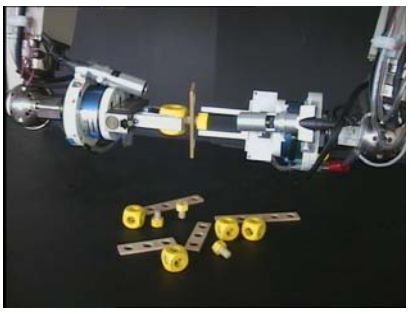

Figure 7: Screwing scenario with two robot hands.

problem that occurred in cooperation tasks with more than one FTS, see [6].

3.3 Analogy

Both problems can be approximately described by Hook's law:

$$\begin{bmatrix} d\vec{x}(t) \\ d\vec{\phi}(t) \end{bmatrix} = \mathcal{C} \begin{bmatrix} \vec{F}(t) \\ \vec{M}(t) \end{bmatrix} \tag{3.1}$$

where $\mathcal{C}$ describes the programmed compliances $d\vec{x}(t)$ and $d\vec{\phi}(t)$ the deviation from the equilibrium position and $\vec{F}(t)$, $\vec{M}(t)$ the forces and torques. They can be learned online from the sensor data of both robots FTS by gradient descent (see [6]):

$$\begin{aligned} \Delta d_{i_1,i_2,\ldots,i_q}(t) &= -\epsilon \frac{\partial E}{\partial d_{i_1,i_2,\ldots,i_q}(t)} \\ &= \epsilon \, (y_r - y_d) \prod_{j=1}^{q} N_{i_j}^{n_j}(x_j), \end{aligned} \tag{3.2}$$

- with the learning rate $0 < \epsilon \leq 1$
- the difference $(y_r - y_d)$ between measured and desired value
- and the B-splines $N_{i_j}^{n_j}(x_j)$

From (3.1) we know that the positional error $(y_r - y_d)$ is approximately proportional to the force error $(F_r - F_d)$.

In the cooperation task every controller has four input dimensions which are the related force and torque, the distance to the desired trajectory and the shoulder–hand distance. All input variables are covered with five B-splines. The rules are extracted as described above.

3.4 Linguistic Rules To Transfer

These rules are to be imported into controllers which are used for a screwing task as discussed in [7]. Because this task is almost static, the third and fourth input variable are dispensable. The screwing operation takes place at a fixed place, so the hand-shoulder distance remains constant. As a consequence the number of rules for controlling each direction is reduced to 25, Tables 1,2 and 3.

NB	NM	NS	NM	NS	NM
NS	NM	NM	NM	NS	NS
ZO	NM	PM	PM	PS	NS
PS	PS	PM	PM	PS	PS
PB	ZO	PS	PS	ZO	PM
	NB	NS	ZO	PS	PB

Table 1: Rules to control forces in x direction.

NB	NM	NM	NM	NM	NM
NS	NB	NM	NM	NM	NB
ZO	PS	ZO	NS	ZO	ZO
PS	PM	PM	PS	ZO	ZO
PB	PM	PM	PM	PM	PM
	NB	NS	ZO	PS	PB

Table 2: Rules to control forces in y direction.

NB	NS	NS	NM	NS	NS
NS	NS	NS	NM	NM	NS
ZO	PM	ZO	NS	NS	NS
PS	PB	PM	ZO	PS	PS
PB	PB	PB	PS	PS	PM
	NB	NS	ZO	PS	PB

Table 3: Rules to control forces in z direction.

Although both tasks (cooperation/screwing) are somehow similar, the stiffness parameters in $\mathcal{C}$ vary a lot. To adapt the different stiffness situations the linguistic terms of the output variable are linearly stretched by multiplication with a stretching factor. If such a factor is unknown, it has to be found iteratively.

3.5 Control with Directly Transferred Rules

Importing these 25 rules in a B-spline controller produces the control surface shown in Fig. 8. Performing a screwing operation like the one in Fig. 9 without online learning will produce the force curves shown in Fig. 10. The desired force in z-direction was -1N (see also Fig. 11 in comparison which shows the forces of an uncontrolled operation[1]). One major benefit of an initialised controller is its faster convergence in relation with learning. Fig. 12 compares the MSE of two controllers, one learning from scratch, the other one after rule transfer. Here one can see how the learning time can be reduced significantly by importing rules from another task.

[1] When such an operation succeeds.

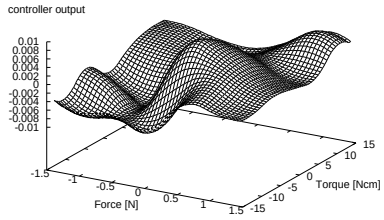

Figure 8: Control surface generated by the imported rules.

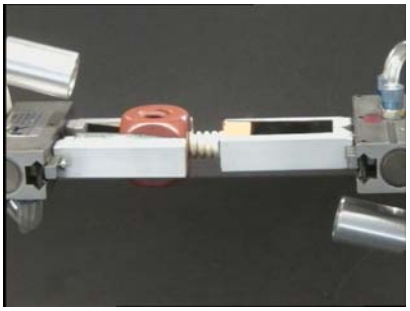

Figure 9: Force guided screwing operation with two robots.

3.6 Optimal Control and Local Adaption

Comparing Fig. 8 and Fig. 13 shows how the learning process modifies the control surface locally but keeps its overall shape. The control surface in Fig. 14 is resulted from a screwing task that was not initialised and trained with seven learning steps. One can see that only the central part of the surface was modified and the outer parts remain in an initial state of zero. This is the problem of bad extrapolation of B-spline controllers which can be handled by a good initialisation.

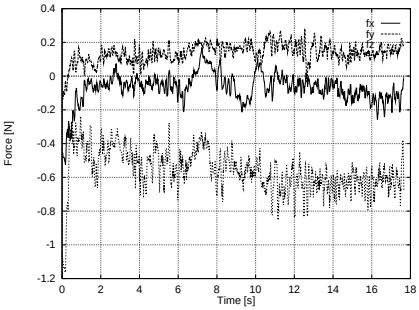

Figure 10: Forces measured during a screwing operation with an initialized controller.

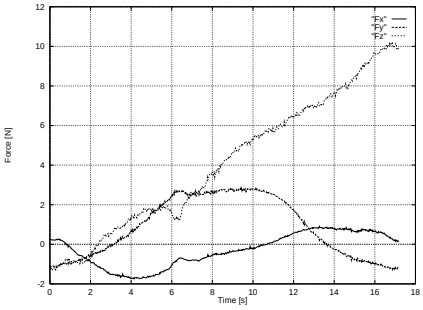

Figure 11: Forces measured during an uncontrolled screwing operation.

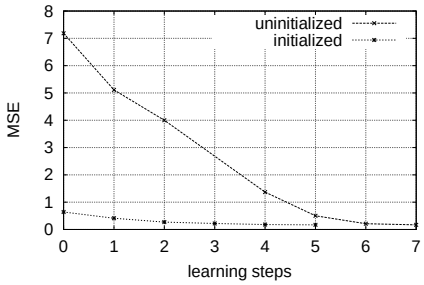

Figure 12: Comparison of the MSE of initialised / uninitialised controllers with respect to the learning steps.

4 Conclusions

We showed that sensor-based behaviours can be incrementally learned based on a B-spline model. Readable rules can be easily extracted from a B-spline fuzzy controller. A set of rules can be transferred to comparable control tasks and locally adapted. Such a skill transfer reduces the learning time significantly. Learning from scratch produces large errors in the early learning process which can prohibit the learning at all (screw does not snatch, ledge breaks, ...). These bootstrapping problems can be avoided.

Several features of the approach are:

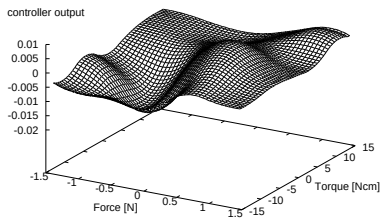

Figure 13: Control surface after learning an initialised controller.

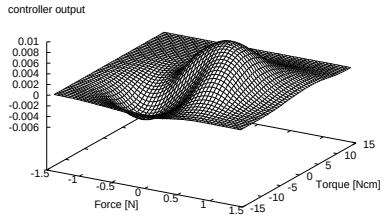

Figure 14: Control surface after learning an uninitialised controller.

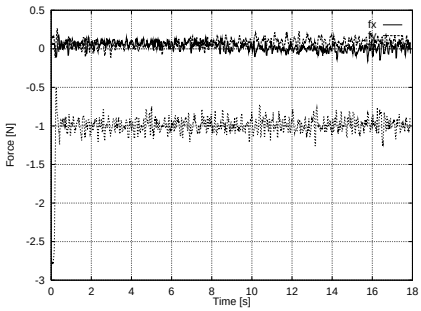

Figure 15: Forces measured during a screwing operation with an initialised and learned controller.

- Knowledge encoding by transforming numerical data to symbolic representation. As a result, huge amounts of data are compressed with the "IF-THEN" structure. If the model of the input/output relation is not available, this compression is quite compact. The proposed model can serve as a bridge between numeric input/output data and symbolic control rules. The approach possesses good interpretability, adaptability and generality if the dimension of the input space is limited.

- Incremental methodology results in the transparency of the behaviour building process. The modular partition of a behaviour in local control rules is actually the reason for rapid convergence of learning.

- Smooth output. If a B-spline basis function of order k is used, the output is $(k-2)$-times continuously differentiable.

With our approach, the perception-action cycle is finally represented in form of "IF-THEN" rules with optimised parameters. No complex programming and control expertise are needed. Fine-tuning of the main controller parameters can be done on-line and automatically. We are working on using this technique for acquiring a wide range of sensor-based skills in a complex aggregate assembly scenario.

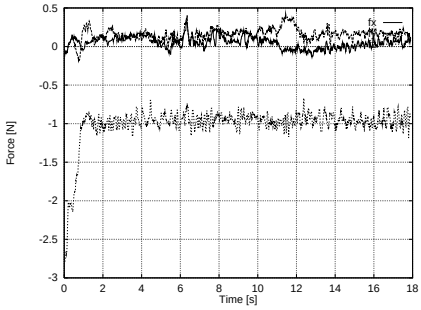

Figure 16: Forces measured during a screwing operation with an uninitialised and learned controller.

References

[1] J. S. Albus. A new approach to manipulator control: The Cerebellar Model Articulation Contorller (CMAC). *Transactions of ASME, Journal of Dynamic Systems Measurement and Control*, 97:220–227, 1975.

[2] W. T. Miller. Real-time application of neural networks for sensor-based control of robots with vision. *IEEE Transactions on System, Man and Cybernetics*, 19:825–831, 1989.

[3] H. Ritter. Parametrized self-organising maps for vision learning tasks. In *ICANN Proceedings*, 1994.

[4] H. Ritter and K. Schulten. Topology conserving mappings for learning motor tasks. In *AIP Conf. Proc. 151, Neural Networks for Computing*, pages 376–380, 1986.

[5] A. Saffiotti, E. H. Ruspini, and K. Konolige. Blending reactivity and goal-directness in a fuzzy controller. *IEEE International Conference on Fuzzy Systems*, pages 134–139, 1993.

[6] J. Zhang and M. Ferch. Rapid On-Line Learning of Compliant Motion for Two-Arm Coordination. *In Proceedings of IEEE International Conference on Robotics and Automation, Leuven,Belgium*, pages 1853 – 1858, 1998.

[7] J. Zhang, Y. Collani and A. Knoll. On-line Learning of B-Spline Fuzzy Controller To Acqire Sensor Based Assembly Skills. *Proceedings of IEEE International Conference on Robotics and Automation, Albuquerque, USA*, pages 1418 – 1423, 1997.

Acknowledgement

This research is funded by the German Research Council (DFG) under grant Graduiertenkolleg "Aufgabenorientierte Kommunikation" and SFB 360-D4.